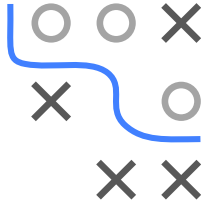
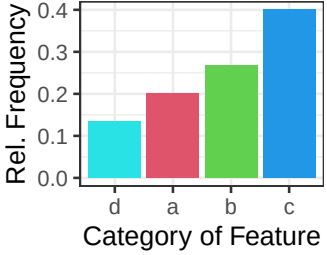
Introduction to Machine Learning



CART
Computational Aspects of Finding Splits

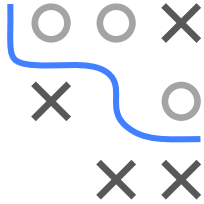


Learning goals

- Know how monotone feature transformations affect the tree
- Understand how categorical features can be treated effectively while growing a CART
- Understand how missing values can be treated in a CART

MONOTONE FEATURE TRANSFORMATIONS

Monotone transformations of one or several features will neither change the value of the splitting criterion nor the structure of the tree, only the numerical value of the split point.

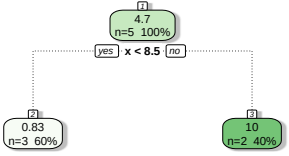


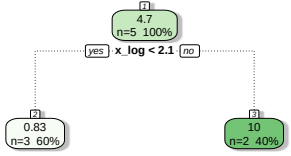
Original data

x	1.0	2.0	7.0	10.0	20.0
y	1.0	1.0	0.5	9.0	11.0

Data with log-transformed x

log(x)	0.0	0.7	1.9	2.3	3.0
y	1.0	1.0	0.5	9.0	11.0

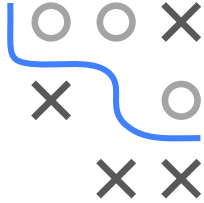


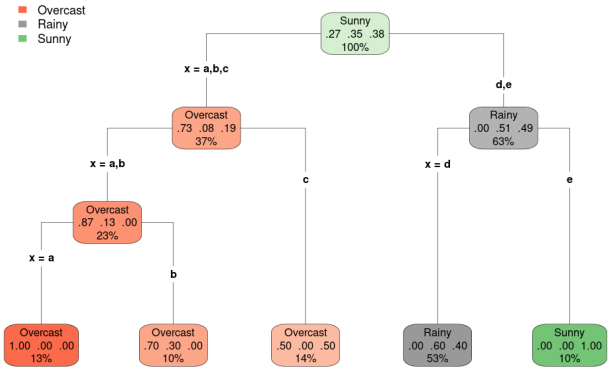


CATEGORICAL FEATURES

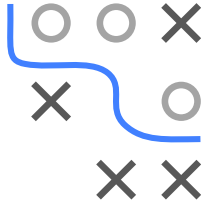
- A split on a categorical feature partitions the feature levels:

$$x_j \in \{a, b, c\} \leftarrow \mathcal{N} \rightarrow x_j \in \{d, e\}$$





CATEGORICAL FEATURES



- A split on a categorical feature partitions the feature levels:

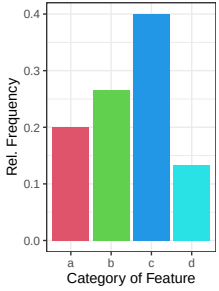
$$x_j \in \{a, b, c\} \leftarrow \mathcal{N} \rightarrow x_j \in \{d, e\}$$

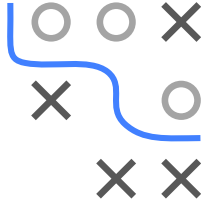
- For a feature with m levels, there are about 2^m different possible partitions of the m values into two groups ($2^{m-1} - 1$ because of symmetry and empty groups).
- Searching over all these becomes prohibitive for large values of m.
- For regression with L2 loss and for binary classification, we can define clever shortcuts.

CATEGORICAL FEATURES

For $0 - 1$ responses, in each node:

1. Calculate the proportion of 1-outcomes for each category of the feature in $\mathcal{N}$.

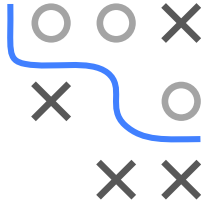


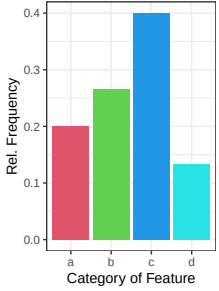


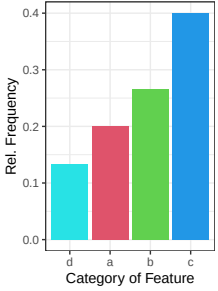
CATEGORICAL FEATURES

For $0-1$ responses, in each node:

1. Calculate the proportion of 1-outcomes for each category of the feature in $\mathcal{N}$.
2. Sort the categories according to these proportions.



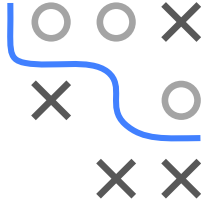


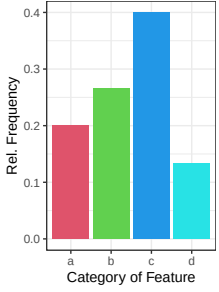


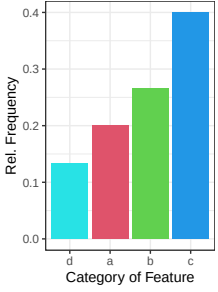
CATEGORICAL FEATURES

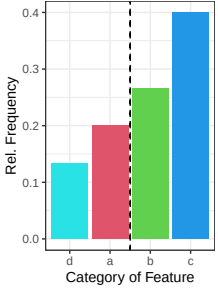
For $0 - 1$ responses, in each node:

1. Calculate the proportion of 1-outcomes for each category of the feature in $\mathcal{N}$.
2. Sort the categories according to these proportions.
3. The feature can then be treated as if it was ordinal, so we only have to investigate at most $m - 1$ splits.

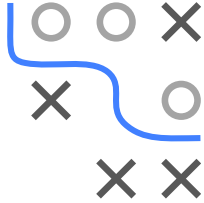








CATEGORICAL FEATURES

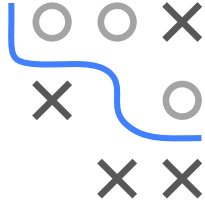


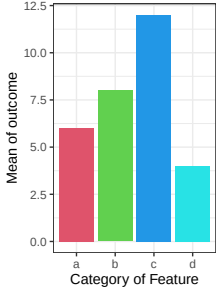
- This procedure finds the optimal split.
- This result also holds for regression trees (with L2 loss) if the levels of the feature are ordered by increasing mean of the target
- The proofs are not trivial and can be found here:
  - for 0-1 responses:
    - ▸ Breiman, 1984, Chapter 4
    - ▸ Ripley, 1996, pp. 213 et seqq.
  - for continuous responses:
    - ▸ Fisher, 1958
- There are only heuristics for the multiclass case ▸ Wright and König, 2019

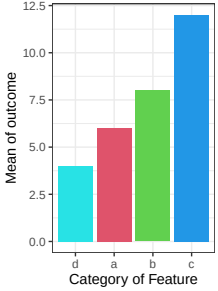
CATEGORICAL FEATURES

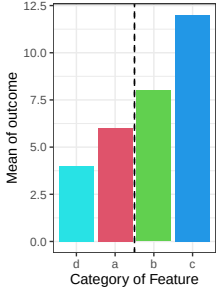
For continuous responses, in each node:

1. Calculate the mean of the outcome in each category
2. Sort the categories by increasing mean of the outcome



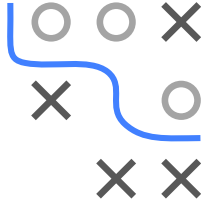




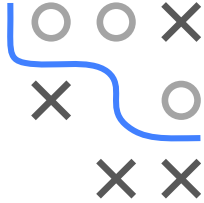


MISSING FEATURE VALUES

- When splits are evaluated, only observations for which the used feature is not missing are used. (This can actually bias splits towards using features with lots of missing values.)
- **Surrogate splits** can deal with missing values during prediction.
- Surrogate splits are created during training. They define replacement splitting rules, using a different feature, that result in almost the same child nodes as the original split.
- When observations are passed down the tree, and the feature value used in a split is missing, we use the surrogate split instead to decide to which child the data should be assigned.



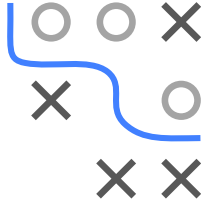
SURROGATE SPLITS



- Each surrogate split is a decision stump that tries to learn the actual splitting rule
- Consider this tree with the primary split w.r.t. `Sepal.Length` where we perform binary classification (`setosa` vs. `virginica`):

- Our surrogate split should optimize a splitting criterion w.r.t. `Sepal.Length < 5.8`

SURROGATE SPLITS



- Consider this subsample of the data used to fit the tree:

	Sepal.Length	...	Petal.Width	Species	Sepal.Length $<$ 5.8
1	5.10	...	0.20	setosa	TRUE
4	4.60	...	0.20	setosa	TRUE
9	4.40	...	0.20	setosa	TRUE
15	5.80	...	0.20	setosa	FALSE
18	5.10	...	0.30	setosa	TRUE
52	5.80	...	1.90	virginica	FALSE
57	4.90	...	1.70	virginica	TRUE
62	6.40	...	1.90	virginica	FALSE
77	6.20	...	1.80	virginica	FALSE
99	6.20	...	2.30	virginica	FALSE

- Add column that indicates whether `Sepal.Length < 5.8`
- Fit tree of depth 1 using all features but `Sepal.Length` to derive a split that explains `Sepal.Length < 5.8` best $\Rightarrow$ surrogate split
- Typically, software stores the best and a few more surrogate splits